Pattern Set Mining with Schema-based Constraint

Pattern Set Mining with Schema-based Constraint

Luca Cagliero, Silvia Chiusano, Paolo Garza*

Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy

Giulia Bruno

Dipartimento di Ingegneria Gestionale e della Produzione, Corso Duca degli Abruzzi 24, 10129, Torino, Italy

Abstract

Pattern set mining entails discovering groups of frequent itemsets that represent potentially relevant knowledge. Global constraints are commonly enforced to focus the analysis on most interesting pattern sets. However, these constraints evaluate and select each pattern set individually based on its itemset characteristics.

This paper extends traditional global constraints by proposing a novel constraint, called schema-based constraint, tailored to relational data. When coping with relational data itemsets consist of sets of items belonging to distinct data attributes, which constitute the itemset schema. The schema-based constraint allows us to effectively combine all the itemsets that are semantically correlated with each other into a unique pattern set, while filtering out those pattern sets covering a mixture of different data facets or

*Corresponding author. Tel.: +39 011 090 7084. Fax: +39 011 090 7099.
Email addresses: luca.cagliero@polito.it (Luca Cagliero), silvia.chiusano@polito.it (Silvia Chiusano), paolo.garza@polito.it (Paolo Garza), giulia.bruno@polito.it (Giulia Bruno)

Preprint submitted to

giving a partial view of a single facet. Specifically, it selects all the pattern sets that are (i) composed only of frequent itemsets with the same schema and (ii) characterized by maximal size among those corresponding to that schema. Since existing approaches are unable to select one representative pattern set per schema in a single extraction, we propose a new Apriori-based algorithm to efficiently mine pattern sets satisfying the schema-based constraint. The experimental results achieved on both real and synthetic datasets demonstrate the efficiency and effectiveness of our approach.
Keywords: Pattern set Mining, Itemset Mining, Data Mining

1. Introduction

Frequent itemsets represent recurrent correlations among data items [1], which are usually selected by considering their local interestingness in the analyzed data [2, 3]. However, since itemset mining from real-life data commonly entails discovering a large number of itemsets that are fairly correlated with each other, the manual inspection of the mining result could be a challenging task. To overcome this issue, pattern set mining with global constraints aims at discovering worthwhile groups of itemsets [4]. Instead of evaluating and selecting itemsets individually, pattern sets (i.e., sets of itemsets) are generated and evaluated as a whole to analyze the correlations among data from a high-level viewpoint.

Relational data is characterized by a fixed schema, which consists of a set of attributes representing peculiar data features. Itemsets mined from relational data are sets of items belonging to distinct data attributes. Hence, they are characterized by a schema too. Frequent itemsets with the same

schema are, to a certain extent, semantically correlated with each other because they are recurrent instances of the same data facet. Hence, the itemset schema can be considered to be particularly suitable for clustering recurrent co-occurrences among data items related to the same facet into pattern sets. Furthermore, instead of generating all the pattern sets complying with a given schema, for each schema only the largest pattern set should be considered, because all the others are partial representations of the same data facet. However, to evaluate pattern set interestingness existing algorithms just evaluate one pattern set at a time. Therefore, they cannot extract for each schema only the best representative pattern set unless generating all the pattern sets first and then postprune the uninteresting ones.

This paper addresses the problem of pattern set mining with global constraints from relational data. To generate only the groups of itemsets containing all the pertinent information related to a given facet, we propose a new global constraint, namely the *schema-based constraint*, tailored to relational data. The schema-based constraint selects all the pattern sets that are (i) composed only of frequent itemsets with the same schema and (ii) characterized by maximal size among those corresponding to that schema. To provide a condensed and potentially useful representation of different data facets we select *at most* one pattern set per schema, i.e., the pattern set that consists of *all and only* the frequent itemsets with that schema.

To improve the manageability of the mined pattern sets two parallel strategies are commonly adopted [4]: (i) enforcing a maximum number of itemsets per pattern set, or (ii) enforcing a minimum percentage of data that must be covered by each mined pattern set. The former constraint, called

cardinality constraint, can be exploited to discard very large and thus unmanageable pattern sets. The latter constraint, named coverage constraint, prevents the extraction of pattern sets representing a small and thus not significant portion of data. Note that our goal is to characterize data using recurrent patterns, rather than pinpointing abnormal (rare) patterns. To efficiently perform pattern set mining with schema-based constraint, we present a new Apriori-based algorithm [5], namely *CO*strained *PA*ttern *S*et mining algorithm (COPAS), which adopts a level-wise approach to discovering itemsets and pattern sets at the same time. The COPAS algorithm pushes the newly proposed schema-based constraint, in conjunction with one of the two traditional constraints (cardinality or coverage, based on users needs), deep into the mining process. In such a way, the pattern sets of interest can be extracted in a single extraction without the need for postprocessing. The result can be directly explored by domain experts for advanced analyses or further processed by using ad hoc strategies.

The paper is organized as follows. Section 2 presents a motivating example. Section 3 compares our work with previous approaches. Section 4 states the mining problem addressed by the paper. Section 5 presents the COPAS algorithm, while Section 6 describes the experiments performed. Finally, Section 7 draws conclusions and discusses future work.

2. Motivating example

A company would like to plan advertising campaigns targeted to customers located in Italy according to their most peculiar features. To personalize advertisements the company clusters customers into segments, which

Table 1: Example relational dataset

Rid	*City*	*Gender*	*Year*	*Job*
1	Turin	F	1980	Teacher
2	Turin	M	1945	Lawyer
3	Turin	M	1945	Lawyer
4	Milan	F	1957	Teacher
5	Rome	M	1976	Clerk
6	Milan	F	1978	Teacher

Table 2: Pattern sets satisfying the schema-based and the minimum coverage constraints mined from the dataset in Table 1 ($minsup$=30%, $mincov$=60%)

Pattern set	Itemsets (support)	Coverage
P_{City}	{(City,Turin)} (50%) {(City,Milan)} (33.3%)	83.3%
P_{Gender}	{(Gender,M)} (50%) {(Gender,F)} (50%)	100%
P_{Job}	{(Job,Teacher)} (50%) {(Job,Lawyer)} (33.3%)	83.3%
$P_{City,Gender}$	{(City,Turin), (Gender,M)} (33.3%) {(City,Milan), (Gender,F)} (33.3%)	66.6%
$P_{City,Job}$	{(City,Turin), (Job,Lawyer)} (33.3%) {(City,Milan), (Job,Teacher)} (33.3%)	66.6%
$P_{Gender,Job}$	{(Gender,F), (Job,Teacher)} (50%) {(Gender,M), (Job,Lawyer)} (33.3%)	83.3%

consist of subsets of customers having similar features. However, deciding the features (or the feature combinations) according to which customers should be clustered is a non-trivial task in large databases.

Table 1 collects some relevant information about the customers under analysis. Each row corresponds to a different customer and it reports the values of a subset of attributes, in particular the city of provenance, gender, year of birth, and job. To achieve their goal, company analysts mine from

the input data itemsets like {(City,Turin), (Gender,M)}, where each itemset is characterized by a given schema (e.g., {City,Gender}). To guarantee itemset relevance, the mined itemsets must hold for at least 30% of the customers, i.e., their frequency of occurrence (support) in the source dataset must be equal to or above a given threshold *minsup*=30%. Then, itemsets with the same schema are analyzed together because they represent the same data facet. For the sake of simplicity, let us consider the itemsets related to pairs of attributes. Since analysts do not know a priori what are the most significant schemata to consider, they have to (i) generate all the itemsets satisfying *minsup*, (ii) cluster the mined itemsets into pattern sets according to their schema, and (iii) rank the pattern sets by decreasing coverage (i.e., the percentage of customers in the dataset for which any itemset in the pattern set holds) and discard those not satisfying a minimum coverage threshold (e.g., *mincov*=60%). At Step (ii) the aforesaid procedure generates 24 pattern sets, because all the possible combinations of the four data attributes are considered. However, only half of them satisfy the coverage constraint and thus they are considered for planning advertising campaigns.

Our approach allows analysts to efficiently extract the subset of pattern sets of interest without generating all the possible itemsets and itemset combinations. Table 2 reports the subset of mined pattern sets. Among the pattern sets related to pairs of attributes, the pattern set with highest coverage is $\{Gender, Job\}$ (83.3%). Each itemset in the pattern represents a combination of customer gender and job, which targets a specific subset of customers. For example, according to customer gender and job, analysts could figure out different advertising policies for female teachers and male

lawyers. Together, the previously mentioned segments cover 83% of the customers thus represent potential targets of advertising campaigns.

3. Related works

Pattern set mining entails discovering groups of itemsets that satisfy a set of global constraints. Instead of selecting patterns based upon their individual merits, global constraints evaluate each pattern set as a whole [6]. Pattern set mining approaches focus on (i) selecting the pattern set that maximizes a certain global quality measure [6, 7, 8, 9, 10, 11, 12, 13, 14] or (ii) discovering all the pattern sets that satisfy a given constraint [4, 15, 16]. Examples of problems related to Task (i) are (a) database tiling [8], which concerns the extraction of the pattern set that covers all the dataset transactions, (b) data compression based on the Minimum Description Length (MDL) principle [12], and (c) pattern set selection by means of constraint programming techniques [9]. Unlike [6, 7, 8, 9, 10, 11, 12, 13, 14], this work addresses the more general Task (ii), i.e., it selects not only the best pattern set but a set of potentially interesting pattern sets.

In [4] the authors formally introduce many different global constraints. Rather than performing pattern set mining as a postprocessing step that follows the traditional itemset mining task [1], in [15, 16] the authors formulate the global constraints directly on the entire itemset space and then accomplish the pattern set mining task using constraint programming techniques. An overview of the constraints used in pattern set mining is given in [4]. For all the previously proposed constraints the selection of a pattern set depends only on the characteristics of its itemsets. Hence, a pattern set cannot be

selected based upon the comparison with other candidate pattern sets. Unlike [4, 15, 16], this paper proposes a new constraint whereby pattern sets are selected not only based upon their own characteristics but also based upon those of other pattern sets. Specifically, the newly proposed schema-based constraint groups itemsets according to their schema and selects, among all the possible pattern sets characterized by the same schema, those having maximal size. In other words, for each schema it selects the pattern set (if any) that contains *all and only* the frequent itemsets with that schema. Since a comparison between pattern sets with the same schema is unfeasible in a single extraction, the mining framework presented in [15] is unsuitable for efficiently addressing pattern set mining problem with schema-based constraint. To avoid generating a large number of pattern sets and then postprune them, we present a new Apriori-based algorithm that pushes the newly proposed schema-based constraint deep into the pattern set mining process.

A large research body has been devoted to mining frequent queries from relational databases [17, 18, 19]. Even though a pattern set with a given schema can be selected by means of group-by query, analysts must (i) know all the attribute combinations in advance and (ii) execute a separate group-by query for each schema, which also enforces support, coverage or cardinality constraints. Since the number of asked queries grows exponentially with the number of attributes, the aforementioned approach is not applicable to large real data.

Parallel research efforts have been devoted to analyzing correlations among data attributes. For example, the authors in [20] introduce the concept of functional dependence. Dependencies are implications among pairs of at-

tribute sets. They indicate that for each dataset record the values assumed by the right-hand side attributes depend on those assumed by the left-hand side ones. Similarly, approximate dependences [21] extend the concept of attribute dependences to the case in which the implications do not hold for all the dataset records. More recently, the authors in [22] focus on discovering sets of low-entropy attributes, i.e., attribute sets for which the corresponding entropy calculated on the projected data is low. Unlike all the aforementioned approaches, this paper focuses on discovering groups of correlations between data items and not between attributes. Even though the concept of low-entropy attribute set implies the existence of recurrent data instances with the same schema, the focus of [22] significantly differs from the one of this work, because we specifically address the problem of pattern set mining under constraints.

The issue of selecting individual itemsets according to their characteristics has already been addressed in [23, 24]. In [23] the authors first formulate the problem of individual itemset mining with constraints. In this context, constraints are conjunctions of boolean predicates which enforce the presence or the absence of a given item combination. Similarly, in [24] an attempt to constrain itemset mining according to the itemset schema in the presence of taxonomies has also been made. Unlike [23, 24] this work focuses on extracting groups of itemsets satisfying global constraints rather than individual itemsets according to their local interestingness. Therefore, the study of the impact of advanced itemset quality measures is out of scope of this work.

4. Pattern set mining with schema-based constraint

This section is organized as follows. First, preliminaries about relational data and itemset mining are given. Secondly, the pattern set mining problem with traditional constraints (i.e., coverage and cardinality constraints) is formulated. Lastly, the integration of schema information into the pattern set mining process is discussed and the problem addressed by the paper is formally stated.

Preliminaries. In the context of relational data, a dataset $\mathcal{R}$ is a set of records and it is characterized by a schema $\Delta=\{\delta_1, \ldots, \delta_n\}$, which consists of a set of attributes $\delta_j \in \Delta$. Each record r, with identifier rid, is a set of items. Item i_j is a pair (δ_j,v_j), where δ_j is an attribute that describes a given data feature, and v_k represents the associated information and belongs to the corresponding attribute domain $dom(\delta_j)$. For the sake of simplicity, hereafter we will not consider datasets that contain null values.

Continuous attribute values are discretized by a preprocessing step. Discretization [25] is commonly applied to real data prior to frequent itemset mining, because continuous values are unlikely to occur frequently in the analyzed data. Although some attempts to mine itemsets and association rules from continuous data have already been made (e.g., [26]), these tasks are out of the scope of our work. On all the analyzed datasets we discretized continuous attributes by using entropy-based discretization [25].

A k-itemset I in $\mathcal{R}$ is a set of k items [1] with distinct attributes, i.e., $I=\{(\delta_1,v_1), \ldots, (\delta_k,v_k)\}$ such that $\delta_j \neq \delta_q \ \forall \ (\delta_j,v_j), (\delta_q,v_q) \in I$. In the following we will denote as $sch(I)$ the *schema* of itemset I, i.e., the set of attributes appearing in I. Itemset I is said to cover a given record $r \in \mathcal{R}$

iff $I \subseteq r$. The ridset of itemset I, denoted as $ridset(I)$, is the set of rids corresponding to the records covered by I in $\mathcal{R}$. The support of I in $\mathcal{R}$ is the percentage of records in $\mathcal{R}$ that are covered by I. If I's support *exceeds* a given threshold *minsup*, then I is said to be *frequent* in $\mathcal{R}$.

For example, I ={(City,Turin), (Gender,M)} is a 2-itemset in the relational dataset in Table 2. Its schema is $sch(I)$={City, Gender}. The support of I in Table 2 is $\frac{2}{6}$, because it covers records with rids 2 and 3, respectively.

In this work we specifically address the extraction of *frequent* itemsets, which are commonly used to characterize large datasets [25]. The complementary issue of discovering abnormal and thus *rare* patterns [27] is out of the scope of this work and it will be addressed as future work.

Pattern set mining set with traditional constraints. Pattern set mining from a relational dataset $\mathcal{R}$ entails discovering subsets of patterns from $\mathcal{R}$ [4].

Definition 1 (Pattern set). *Let $\mathcal{I}$ be the set of all itemsets in a relational dataset $\mathcal{R}$. $P \subseteq \mathcal{I}$ is a pattern set in $\mathcal{R}$.*

Hereafter we will denote by $\mathcal{P}$ the set of all the possible pattern sets in a relational dataset $\mathcal{R}$, i.e., $\mathcal{P}$=$2^{\mathcal{I}}$. Since this work focuses on discovering interesting *sets of itemsets*, the individual patterns occurring in a pattern set will be denoted as *itemsets* throughout the paper.

Pattern sets are characterized by different quality measures [4]. Hereafter we will consider two traditional quality measures, namely the set *cardinality* and *coverage*, which deemed as particularly suitable for making the pattern sets manageable by domain experts for manual inspection [4]. Their formal definitions are given below.

Definition 2 (Pattern set cardinality and coverage). *Let $P_i \in \mathcal{P}$ be an arbitrary pattern set. The definitions follow.*

(i) The cardinality of P_i, denoted as $card(P_i)$, is the number of itemsets in P_i, i.e, $card(P_i) = |P_i|$.

(ii) The coverage of P_i, denoted as $cov(P_i)$, is the percentage of records in $\mathcal{R}$ covered by any itemset in P_i, i.e.,

$$cov(P_i) = \frac{|\{r \in \mathcal{R} |\ \exists\ I \in P_i\ s.t.\ I \subseteq r\}|}{|\mathcal{R}|} = \frac{|\bigcup_{I \in P_i} ridset(I)|}{|\mathcal{R}|} \quad (1)$$

Let us consider again the example pattern sets reported in Table 2. Pattern set $P_{City,Gender}$ has cardinality equal to 2 and coverage equal to 66.6%. It contains itemsets {(City,Turin), (Gender,M)} and {(City,Milan), (Gender,F)}, which cover the records with rids $\{2, 3\}$ and $\{4, 6\}$, respectively. Hence, the coverage of the pattern set is 66%.

The cardinality measure evaluates pattern set handiness. Groups composed of few itemsets (i.e., low-cardinality pattern sets) are typically more easily manageable by domain experts for manual result inspection than high-cardinality ones. The coverage measure indicates how groups are representative of the analyzed data. For instance, pattern sets that cover many records (i.e., high-coverage pattern sets) characterize larger amounts of data than low-coverage sets. Since the goal of this work is to characterize recurrent patterns rather than pinpointing abnormal or unexpected behavior, high-coverage sets are, in general, deemed as more actionable than low-coverage ones for advanced analyses.

Minimum coverage and maximum cardinality constraints are global constraints that are commonly enforced to pick out the most relevant pattern

sets [4]. They select the pattern sets that represent a large enough portion of the analyzed data (i.e., $cov(P_i) \geq mincov$) and that have manageable size (i.e., $card(P_i) \leq maxcard$), respectively.

Integrating schema information into the pattern set mining process. This paper investigates the use of the pattern set schema to select potentially interesting pattern sets. According to the itemset schema, pattern sets can be classified as follows: (i) pattern sets containing all the itemsets with a given schema, (ii) pattern sets containing a subset of the itemsets with a given schema, or (iii) pattern sets containing a mixture of itemsets with different schema. Frequent itemsets with the same schema are, to a certain extent, semantically correlated with each other because they are recurrent instances of the same data facet. Hence, we are interested in discarding those pattern sets that contain a mixture of different schemata (type (iii)) because they are not targeted to any specific data facet. On the other hand, to have a global view on a given facet we would like to combine in a single pattern set all the itemsets with the corresponding schema. Therefore, we would like to extract only the itemsets of type (i). To achieve our goal, we introduce a new global constraint, called *schema-based constraint*.

Pattern set mining with schema-based constraint selects, among the pattern sets that contain only itemsets with the same schema, at most one representative pattern set per schema. A more formal definition follows.

Definition 3 (Schema-based constraint). *Let minsup be a minimum support threshold and let $\mathcal{I}_f$ be the set of frequent itemsets in a relational dataset $\mathcal{R}$ according to minsup. An arbitrary pattern set $P_i \in \mathcal{P}$ satisfies the schema-based constraint if and only if:*

(i) P_i *contains* ***only*** *frequent itemsets in* $\mathcal{I}_f$ *with the same schema, i.e.,* $\forall\ I_j,\ I_q \in P_i,\ sch(I_j)=sch(I_q)$

(iii) P_i *contains* ***all*** *the frequent itemsets in* $\mathcal{I}_f$ *with the same schema, i.e.,* $\forall\ I_j, I_q \in \mathcal{I}_f$ *such that* $sch(I_j) = sch(I_q)$ *then* $I_j, I_q \in P_i$.

Given a pattern set P_i satisfying the schema-based constraint, with convenient abuse of notation we will denote by *pattern set schema* $sch(P_i)$ the schema of any itemset in P_i throughout the paper.

The pattern set mining problem with schema-based constraint is a new and challenging task, because it cannot be neither reformulated as a combination of the previously proposed global constraints nor efficiently tackled with state-of-the-art pattern set mining frameworks (e.g., [15]). A more detailed comparison with the state-of-the-art is given in Section 3.

To generate interesting and manageable pattern sets, we focus on combining the schema-based constraint with traditional pattern set mining constraints.

Problem statement.

Let $\mathcal{R}$ be a relational dataset, *minsup* a minimum support threshold, and C a traditional global constraint, either the minimum coverage or the maximum cardinality constraint. The pattern set mining problem with schema-based constraint entails the extraction from $\mathcal{R}$ of all the pattern sets that are composed of frequent itemsets and that satisfy (i) the schema-based constraint and (ii) one global constraint C (of user's choice between cardinality and coverage).

Table 2 reports the pattern sets mined from the dataset in Table 1 by enforcing the schema-based and minimum coverage constraints.

While tackling the pattern set mining problem with schema-based constraint from relational data, both minimum coverage and maximum cardinality constraints satisfy the anti-monotonicity property.

Property 1. *(Anti-monotonicity property of the minimum coverage and maximum cardinality constraints): Let $P_i, P_j \in \mathcal{P}$ be two arbitrary pattern sets that satisfy the schema-based constraint. Let $\prec$ be a generality relation such that $P_i \prec P_j$ iff $sch(P_i) \subseteq sch(P_j)$. Let mincov be a minimum coverage threshold and maxcard a maximum cardinality threshold. The following properties hold:*

(i) The minimum coverage constraint $cov(P) \geq mincov$ is anti-monotone w.r.t. $\prec$, i.e., if $cov(P_i) < mincov$ then $cov(P_j) < mincov$.

(ii) The maximum cardinality constraint $card(P) \leq maxcard$ is anti-monotone w.r.t. $\prec$, i.e., if $card(P_i) > maxcard$ then $card(P_j) > maxcard$, if minsup=0 is enforced.

Proof 1 (Proof of Property (i)). *Let P_i, $P_j \in \mathcal{P}$ be two arbitrary pattern sets satisfying the schema-based constraint such that $P_i \prec P_j$. We would like to prove that $cov(P_j) \leq cov(P_i)$. Since $P_i \prec P_j$, then $sch(P_i) \subseteq sch(P_j)$. Furthermore, due to the schema-based constraint, $\forall\ I_i \in P_i\ sch(I_i) = sch(P_i)$, and $\forall\ I_j \in P_j\ sch(I_j) = sch(P_j)$. Since $sch(P_i) \subseteq sch(P_j)$ it follows that $sch(I_i) \subseteq sch(I_j)$. Without any loss of generality, let us consider $sch(P_i) = \{\delta_1, \ldots, \delta_k\}$ and $sch(P_j) = \{\delta_1, \ldots, \delta_k, \delta_{k+1}\}$, $\delta_q \in \Delta\ 1 \leq q \leq k+1$. Given an arbitrary itemset $I_j \in P_j$, let $I_i \subset I_j$ be the itemset generalization of I_j obtained by removing item (δ_{k+1}, v_{k+1}), $v_{k+1} \in dom(\delta_{k+1})$, from I_j. Due to the anti-monotonicity property of the support measure [5], $ridset(I_j) \subseteq$*

ridset(I_i). *Given that* $sch(I_i) = sch(P_i)$ *and* I_i *is frequent because* I_i ($I_j \subseteq I_i$) *is frequent too, then* $I_i \in P_i$. *Thus, it follows that* $\bigcup_{I_j \in P_j} ridset(I_j) \subseteq \bigcup_{I_i \in P_i} ridset(I_i)$. *Therefore, by Definition 2, the inequality* $cov(P_j) \leq cov(P_i)$ *holds.*

Proof 2 (Proof of Property (ii)). *Let* P_i, $P_j \in \mathcal{P}$ *be two arbitrary pattern sets satisfying the schema-based constraint such that* $P_i \prec P_j$. *We would like to prove that* $card(P_j) \geq card(P_i)$. *Since* $P_i \prec P_j$, *then* $sch(P_i) \subseteq sch(P_j)$. *Moreover, due to the schema-based constraint,* $\forall\ I_i \in P_i\ sch(I_i)$=$sch(P_i)$, *and* $\forall\ I_j \in P_j\ sch(I_j) = sch(P_j)$. *Since* $sch(P_i) \subseteq sch(P_j)$ *it follows that* $sch(I_i) \subseteq sch(I_j)$. *Without any loss of generality, let us consider* $sch(P_i) = \{\delta_1, \ldots, \delta_k\}$ *and* $sch(P_j) = \{\delta_1, \ldots, \delta_k, \delta_{k+1}\}$, $\delta_q \in \Delta\ 1 \leq q \leq k+1$. *Given an arbitrary itemset* $I_i \in P_i$, *let* $\mathcal{I}_j$=$\{I_{j_1}, \ldots, I_{j_m}\}$ *be the set of itemsets in* $\mathcal{R}$ *such that* $\forall\ I_{j_p} \in \mathcal{I}_j$, $sch(I_{j_p})$=$sch(P_j)$ *and* $I_i \subset I_{j_p}$, *i.e.,* I_{j_p} *is an itemset specialization of* I_i *with schema* $sch(P_j)$. *Each itemset* $I_{j_p} \in \mathcal{I}_j$ *is obtained by adding a different item with attribute* δ_{k+1} *to* I_i, *i.e.,* $I_{j_p} = I_i \cup \{(\delta_{k+1}, v_{k+1_p})\}$, $v_{k+1_p} \in dom(\delta_{k+1})$. *For the sake of readability, the rest of the proof is divided into two steps.*

1. *Since, by construction, the null value is not allowed for* δ_{k+1}, *then* $\mathcal{I}_j \neq \emptyset$. *Furthermore, since* $sch(I_{j_p}) = sch(P_j)$ *and* $minsup = 0$, *then it follows that all itemsets* $I_{j_p} \in \mathcal{I}_j$ *are contained in* P_j.
2. *Let* $I_{i_1}, I_{i_2} \in P_i$, $I_{i_1} \neq I_{i_2}$ *be two arbitrary itemsets. Let* $\mathcal{I}_{j_1}$ *and* $\mathcal{I}_{j_2}$ *be the set of itemset specializations of* I_{i_1} *and* I_{i_2}, *respectively. Since* $I_{i_1} \neq I_{i_2}$ *then* $\forall\ I_{j_{1_p}} \in \mathcal{I}_{j_1}$, $\forall\ I_{j_{2_q}} \in \mathcal{I}_{j_2}$, $I_{j_{1_p}} \neq I_{j_{2_q}}$.

Thanks to (1), $\mathcal{I}_{j_1}$ *and* $\mathcal{I}_{j_2}$ *defined as in (2) are both contained in* P_j. *Therefore, combining (1) with (2), it follows that* $card(P_j) \geq card(P_i)$.

As discussed in the following section, the aforesaid properties allow us to push the schema-based constraint in conjunction with one traditional constraint (either cardinality or coverage) deep into the mining process.

5. The COPAS algorithm

Algorithm 1 The COPAS algorithm

Input: relational dataset $\mathcal{R}$, minimum support threshold *minsup*, global constraint C
Output: pattern sets satisfying the schema-based and global constraint C

1: $k = 1$
2: $\mathcal{FI}_k$=generate-all-frequent-1-itemsets(*minsup*)
3: $\overline{\mathcal{SP}}_k$=itemset-grouping($\mathcal{FI}_k$) */* Generate the candidate pattern sets composed of 1-itemsets satisfying the schema-based constraint */*
4: **for** each candidate pattern set $P \in \overline{\mathcal{SP}}_k$ **do**
5: compute-pattern-set-measures(P) */* Compute the quality measures for the candidate pattern set P */*
6: **end for**
7: $\mathcal{SP}_k$ = apply-global-constraint($\overline{\mathcal{SP}}_k$, C) */* Remove the pattern sets not satisfying the global constraint C */*
8: **while** $\mathcal{SP}_k \neq \emptyset$ **do**
9: $k = k + 1$
10: $\widehat{\mathcal{SP}}_k$ = generate-candidate-pattern-sets-and-itemsets($\mathcal{SP}_{k-1}$) */* Generate the candidate pattern sets that satisfy the schema-based constraint along with their candidate itemsets */*
11: **for** each candidate pattern set $P \in \widehat{\mathcal{SP}}_k$ **do**
12: compute-itemset-support(P) */* Compute support for candidate itemsets in pattern set P */*
13: **end for**
14: $\overline{\mathcal{SP}}_k$ = apply-support-constraint($\widehat{\mathcal{SP}}_k$,*minsup*) */* Remove the infrequent itemsets from the candidate pattern sets */*
15: **for** each candidate pattern set $P \in \overline{\mathcal{SP}}_k$ **do**
16: compute-pattern-set-measures(P) */* Compute measures for candidate pattern set P */*
17: **end for**
18: $\mathcal{SP}_k$ = apply-global-constraint($\overline{\mathcal{SP}}_k$, C) */* Remove the pattern sets not satisfying the global constraint C */*
19: **end while**
20: **return** $\cup_k \mathcal{SP}_k$

COPAS is an Apriori-based [5] algorithm which tackles the pattern set mining problem stated in Section 4. To accomplish the mining task efficiently, it generates itemsets of increasing length, along with their corresponding pattern sets, in a level-wise manner. At an arbitrary k-th step, the candidate k-itemsets and their corresponding pattern sets are generated first. Then, the infrequent itemsets and candidate pattern sets that do not satisfy the

constraints are pruned. Thanks to the anti-monotonicity property of the coverage and cardinality constraints (see Property 1), only the itemsets and pattern sets that were selected at the k-th iteration are used at the $(k+1)$-th iteration to generate the itemsets and their corresponding pattern sets.

Algorithm 1 reports the COPAS pseudo-code. Firstly, the frequent 1-itemsets are selected and partitioned into pattern sets with schema of length 1 (lines 1-7). Then, the pattern sets that do not satisfy the global constraints are discarded. Next, an iterative procedure is triggered. An arbitrary k-th iteration ($k \geq 2$) entails the following steps:

(i) *Candidate itemset and pattern set generation.* The procedure generates the k-itemsets and their corresponding pattern sets at the same time (line 10). Any pattern set that satisfies the schema-based constraint must contain only itemsets with the same schema. Hence, to generate a candidate pattern set that contains k-itemsets and satisfies the schema-based constraint COPAS joins pairs of pattern sets containing $(k-1)$-itemsets. Such pattern sets are generated at the $(k-1)$-th iteration and collected into set $\mathcal{SP}_{(k-1)}$ (line 18). More specifically, for each pattern set $P_i \in \mathcal{SP}_{(k-1)}$ the schema attributes are sorted in lexicographical order. Furthermore, the items contained in each itemset $I_i \in P_i$ are sorted in the same way. Two pattern sets $P_i, P_j \in \mathcal{SP}_{(k-1)}$ are joined if they share the first $(k-2)$ schema attributes. The resulting pattern set P_t has schema $sch(P_i) \cup sch(P_j)$ and it contains the itemsets generated by joining the itemsets in P_i and P_j. Similar to Apriori [5], for each pair of itemsets $I_i, I_j \in P_i$ sharing the first $(k-2)$ items, the corresponding itemset $I_t{=}I_i \cup I_j \in P_t$ is generated.

(ii) *Itemset and pattern set evaluation and selection.* The support of each

candidate k-itemset is computed by performing a dataset scan and the infrequent itemsets are discarded (lines 11-14). Furthermore, the pattern sets satisfying the global constraint C, i.e., the coverage or the cardinality constraint, are selected (lines 15-18). Since distinct itemsets with the same schema cannot cover the same record, the coverage of a pattern set can be straightforwardly computed by summing the support values of its itemsets.

The iterative procedure stops when no further candidate pattern set is generated, i.e., when $\mathcal{SP}_k$ becomes empty (line 8). The mining result contains all the pattern sets satisfying the constraints as well as their corresponding frequent itemsets.

As an example, let us consider the dataset in Table 1 and the patterns in Table 2 mined by enforcing *minsup*=33% and *mincov*=50%. COPAS first generates pattern sets P_{City}, P_{Gender}, P_{Year}, and P_{Job}. P_{Year} is discarded, along with the corresponding itemset $\{(Year, 1945)\}$, because it does not satisfy the *mincov* constraint. The remaining sets are added to the output set and they are used to generate candidate sets $P_{City,Gender}$, $P_{City,Job}$ and $P_{Gender,Job}$. For each pair of joined pattern sets, their corresponding itemsets are joined. For example, pattern set $P_{City,Gender}$ contains 4 candidate itemsets generated by joining the itemsets in P_{City} and P_{Gender}. Among them, only {(City,Turin),(Gender,M)} and {(City,Milan),(Gender,F)} are frequent and, thus, they are included in $P_{City,Gender}$. Pattern sets $P_{City,Gender}$, $P_{City,Job}$, and $P_{Gender,Job}$ are selected because they satisfy the *mincov* constraint. Finally, at the third iteration, $P_{City,Gender}$ and $P_{City,Job}$ are joined because they share the first schema attribute (i.e., $City$) and the corresponding pattern set $P_{City,Gender,Job}$ is generated. At the same time, frequent 3-itemsets

Table 3: Characteristics of the UCI datasets analyzed

Dataset	Number of records	Number of attributes	Estimated density
Adult	30,162	15	6.58
Letter-rec.	20,000	17	1.55
Mushroom	8,124	23	6.88
Pendigits	10,992	17	1.48
Poker	1,025,010	11	2.89
Shuttle	58,000	10	3.34
Vehicle	894	19	2.31
Voting	435	17	2.11
Waveform	5,000	22	1.54

{(City,Turin),(Gender,M),(Job,Lawyer)} and {(City,Milan),(Gender,F),(Job,Teacher)} are generated and included in $P_{City,Gender,Job}$.

6. Experiments

We performed an extensive set of experiments on real and synthetic datasets to analyze

1. The *usefulness* of the proposed approach in a real application context (Section 6.2).
2. The *selectivity* of the newly proposed schema-based constraint, in association with coverage and cardinality constraints and not (Section 6.3).
3. The *execution time* the COPAS algorithm compared to those of existing approaches (Section 6.5), and
4. The *scalability* of the newly proposed COPAS algorithm with the number of dataset records and attributes (Section 6.6).

The experiments related to Tasks (1) and (2) were run on a subset of representative UCI relational datasets [28], whereas the algorithm scalability

was evaluated on synthetic data. Table 3 summarizes the main UCI dataset characteristics, where we considered the density measure described in [29][1].

The experiments were performed on a 2.8-GHz Intel Pentium IV PC with 2 GBytes of main memory, running Ubuntu 10.04. COPAS was developed in C and its executable code is available at [30].

6.1. State-of-the-art competitors

To the best of our knowledge, state-of-the-art pattern set mining algorithms (e.g., [4, 15]) are unable to extract only the pattern sets satisfying the schema-based constraint in a single extraction, because they do not allow us to select at most one pattern set per schema. Hence, in general, the pattern sets mined by the COPAS algorithm are not directly comparable with those generated by the other algorithms unless applying a postprocessing step.

To compare the performance of the COPAS algorithm with that of state-of-the-art algorithms we integrated the existing algorithms in a two-step process: (i) candidate pattern set and itemset generation using traditional algorithms and (ii) pattern set pruning driven by constraints. Step (i) entails extracting a superset of the pattern sets mined by COPAS by using traditional algorithms, while step (ii) discards the pattern sets not satisfying the schema-based constraint.

We considered two complementary strategies. The first one, namely POST-FPMINE, performs FP-growth-like itemset mining [31] followed by pattern set generation and postpruning. The second one, namely POST-CPMINE, performs pattern set mining using an established constraint programming-

[1]The dataset density is defined as the average local support of the FP-Tree nodes [29] in the FP-tree data structure representing the entire dataset (i.e., setting *minsup* to 0).

based approach [15], and then it filters out the pattern sets not satisfying the schema-based constraint. Since the COPAS algorithm relies on an Apriori-based itemset mining algorithm, testing the POST-FPMINE strategy is deemed as interesting to evaluate the effectiveness of pushing the schema-based constraint deep into the mining process. Note that, although Apriori is known to be less scalable than FP-Growth on dense datasets [31], it allows us to prevent the extraction of some uninteresting pattern sets thanks to the anti-monotonicity property of the coverage/cardinality constraints (see Section 4). On the other hand, the POST-CPMINE strategy relies, to the best of our knowledge, on the most recent and efficient state-of-the-art pattern set mining algorithm [15]. Hence, comparing the performance of the COPAS algorithm with that of POST-CPMINE allows us to evaluate the efficiency of the proposed approach against the state-of-the-art. Even though the intermediate results of the POST-FPMINE and POST-CPMINE strategies are different, their outputs correspond to those of the COPAS algorithm. A more detailed description of the POST-FPMINE and POST-CPMINE mining strategies is given below.

POST-FPMINE. The POST-FPMINE algorithm consists of three separate steps:

(i) *Frequent itemset extraction. All* frequent itemsets are extracted from the input dataset using the established FP-growth algorithm [31].

(ii) *Candidate pattern set computation.* Frequent itemsets are partitioned in pattern sets based on their schema by enforcing the schema-based constraint and the measures (i.e., cardinality and coverage) characterizing each pattern set are computed.

(iii) *Pattern set selection.* The global pattern set constraint (either the cardinality or the coverage constraint) is applied and the pattern sets not satisfying the enforced constraint are discarded.

POST-CPMINE. The POST-CPMINE algorithm consists of three separate steps:

(i) *Extraction of pattern sets composed of frequent itemsets.* All the pattern sets that are exclusively composed of frequent itemsets are extracted by using the publicly available implementation of the approach proposed in [15].

(ii) *Schema-based constraint enforcement.* The subset of pattern sets that satisfy the schema-based constraint are selected (i.e., for each schema the pattern set that contains all of the frequent itemsets with that schema is selected).

(iii) *Cardinality/Coverage constraint enforcement.* The subset of pattern sets satisfying also the cardinality/coverage constraint are selected.

As discussed in Section 6.5, the main drawback of the POST-FPMINE and POST-CPMINE strategies is that they unnecessarily generate a large number of itemsets and pattern sets at Step (1) and (2), which are then pruned at Step (3). Note that, when dealing with complex datasets, the intermediate results may not fit in main memory.

6.2. Result validation

We evaluated the usability of the proposed approach for planning marketing campaigns based on the analysis of real census data. To perform our

analyses, we considered the UCI benchmark dataset Adult [28], which consists of demographic data about 30,162 American persons. For each person the dataset also contains the information about the annual compensation (less than or above 50K USD).

To plan personalized advertising campaigns, marketing officers are commonly interested in analyzing customer data to segment customers according to their most peculiar demographic and economic features. However, identifying the most appropriate customer segments is a challenging task, because analysts should first analyze many different data facets (e.g., age, gender, city of provenance, compensation) at the same time, and then, for each combination of data facets, they have to evaluate the correlation between the corresponding values.

For example, officers may wonder what is the most appropriate subset of features to perform customer segmentation. To answer this question they should analyze all the possible data segmentations (i.e., 2^{15}=32,768 attribute combinations on Adult). Next, for each segmentation they have to figure out what are the most appropriate advertising rules to apply. For example, to advertise luxury products based on customer age and compensation they may wonder what are the customer age groups that are strongly correlated with a compensation above 50K USD.

For each combination of customer facets the COPAS algorithm generates at most one pattern set, where each pattern set contains all the corresponding customer segments.

Table 4 reports the pattern sets mined from Adult by enforcing a minimum coverage equal to 95%. For example, pattern set $P_{Gender,Income}$ segments

Table 4: Adult: pattern sets mined enforcing $mincov = 95\%$ and $minsup = 1\%$

Pattern set	Cardinality	Coverage
P_{Income}	2	100
$P_{Gender,Income}$	4	100
$P_{Workclass,Income}$	12	100.0
$P_{Age,Income}$	12	98.4
$P_{Capital-gain,Income}$	4	97.3
$P_{Capital-loss,Income}$	3	96.8
$P_{Education,Income}$	17	95.2
$P_{Education-num,Income}$	12	98.5
$P_{Marital-status,Income}$	9	99.3
$P_{Occupation,Income}$	18	96.0
$P_{Hours-per-week,Income}$	10	100
$P_{Race,Income}$	5	97.5
$P_{Relationship,Income}$	9	99.0
$P_{Age,Gender,Income}$	22	97.6
$P_{Education,Education-num,Income}$	17	95.2
$P_{Education-num,Gender,Income}$	19	95.9
$P_{Gender,Capital-gain,Income}$	5	95.2
$P_{Gender,Capital-loss,Income}$	4	95.7
$P_{Marital-status,capital-loss,Income}$	9	95.1
$P_{Marital-status,Gender,Income}$	12	95.8
$P_{Race,Gender,Income}$	7	95.4
$P_{Relationship,Gender,Income}$	13	98.0
$P_{Workclass,Capital-loss,Income}$	12	95.7
$P_{Workclass,Gender,Income}$	16	96.6
$P_{Gender,Hours-per-week,Income}$	14	96.6
...	...	...

customers according to gender (male or female) and compensation ($\leq$50K, >50K). It consists of four frequent itemsets, each one representing a different customer segment (e.g., males who earn more than 50K USD). Since customer age, gender, and yearly income are commonly used to profile customer preferences, the domain expert suggests us to consider the following combinations of data facets: (i) gender and income, and (ii) gender, income, and age. Table 5 reports the itemsets related to pattern sets $P_{Gender,Income}$ and $P_{Age,Gender,Income}$. Note that only the itemsets that hold for 1% of the customers are considered because planning ad-hoc campaigns targeted to small segments is not worthy.

Let us consider first the itemsets belonging to pattern set $P_{Gender,Income}$. Each itemset represents a disjoint segment covering at least 3.7% of the original customers. From the extracted patterns it appears that (i) the majority of the customers is male and not wealthy (compensation $\leq$ 50 USD). (ii) most of the wealthy customers (compensation $\geq$ 50 USD) is male (21.2% male vs. 3.7% female). Therefore, targeting promotions of luxury goods to males appears to be convenient. However, regardless of the customer gender, non-luxury goods have a broader target than luxury ones (a campaign targeted to non-wealthy customers reaches 75% of the customers).

To deepen the analysis officers may further segment customers according to their age group by exploiting the information provided by pattern set $P_{Age,Gender,Income}$, which is a specialization of $P_{Gender,Income}$. The latter pattern set contains 22 frequent itemsets, which represent different and potentially interesting customer segments. A manual inspection of the pattern set allows marketing officers to plan finer promotions without the need for

Table 5: Adult: content of the pattern sets $P_{Gender,Income}$ and $P_{Age,Gender,Income}$ mined enforcing $mincov = 95\%$ and $minsup = 1\%$

Pattern set	Coverage	Itemsets	support
$P_{Gender,Income}$	100%	{(Gender,Female), (Income,>50K)}	3.7%
		{(Gender,Female), (Income,≤50K)}	28.7%
		{(Gender,Male), (Income,>50K)}	21.2%
		{(Gender,Male), (Income,≤50K)}	46.4%
$P_{Age,Gender,Income}$	97.6%	{(Age,< 21.5), (Gender,Female), (Income≤50K)}	4.1 %
		{(Age, < 21.5), (Gender, Male), (Income ≤50K)}	4.6 %
		{(Age, [21.5-23.5)), (Gender, Female), (Income ≤50K)}	2.1 %
		{(Age, [21.5-23.5)), (Gender, Male), (Income ≤50K)}	2.8 %
		{(Age, [23.5-27.5)), (Gender, Female), (Income ≤50K)}	3.5 %
		{(Age, [23.5-27.5)), (Gender, Male), (Income ≤50K)}	6.0 %
		{(Age, [27.5-29.5)), (Gender, Female), (Income ≤50K)}	1.5 %
		{(Age, [27.5-29.5)), (Gender, Male), (Income ≤50K)}	2.9 %
		{(Age, [29.5-35.5)), (Gender, Female), (Income ≤50K)}	4.2 %
		{(Age, [29.5-35.5)), (Gender, Male), (Income ≤50K)}	8.4 %
		{(Age, [35.5-43.5)), (Gender, Female), (Income ≤50K)}	5.0 %
		{(Age, [35.5-43.5)), (Gender, Male), (Income ≤50K)}	8.7 %
		{(Age, [43.5-61.5)), (Gender, Female), (Income ≤50K)}	6.8 %
		{(Age, [43.5-61.5)), (Gender, Male), (Income ≤50K)}	10.6 %
		{(Age, ≥61.5), (Gender, Female), (Income ≤50K)}	1.6 %
		{(Age, ≥61.5), (Gender, Male), (Income ≤50K)}	2.3 %
		{(Age, [29.5-35.5)), (Gender, Male), (Income >50K)}	3.2 %
		{(Age, [35.5-43.5)), (Gender, Female), (Income >50K)}	1.2 %
		{(Age, [35.5-43.5)), (Gender, Male), (Income >50K)}	5.9 %
		{(Age, [43.5-61.5)), (Gender, Female), (Income >50K)}	1.3 %
		{(Age, [43.5-61.5)), (Gender, Male), (Income >50K)}	9.8 %
		{(Age, ≥61.5), (Gender, Male), (Income >50K)}	1.1 %

Table 6: Effect of the schema-based constraint

Dataset	*minsup*(%)	Number of pattern sets	Number of freq. itemsets	Avg. number of freq. itemsets per pat. set	Avg. coverage (%) per pat. set
Adult	0	32,767	1.00E+08	3052	100
	0.05	32,767	9.00E+06	275	67.30
	0.1	32,767	4.00E+06	122	58.06
	1	27,689	3.00E+05	12	28.79
Letter-rec.	0	131,071	1.00E+09	7629	100
	0.025	131,071	4.00E+07	305	14.77
	0.035	131,071	2.00E+07	153	10.41
	1	2,794	1.00E+04	5	8.98
Mushroom	0	8,388,607	1.32E+09	157	100
	1	4,350,279	9.13E+07	21	41.17
	1.5	3,370,551	4.80E+07	14	37.27
	2	2,525,235	2.40E+07	10	32.85
Pendigits	0	131,071	1.10E+09	8392	100
	0.1	116,488	5.10E+06	44	7.26
	0.5	87,199	1.90E+06	22	7.07
	1	5,798	2.00E+04	4	5.27
Poker	0	2,048	4.00E+08	200317	100
	0.5	171	9.00E+03	51	98.30
	0.75	111	6.00E+03	53	92.36
	1	76	2.70E+03	36	69.92
Shuttle	0	1,023	2.00E+07	19550	100
	0.01	963	1.00E+06	1038	59.38
	0.015	951	9.00E+05	946	53.26
	1	200	2.00E+03	12	28.36
Vehicle	0	524,287	1.70E+08	324	100
	0.5	524,287	2.60E+07	50	55.21
	1	524,287	9.90E+06	19	36.76
	1.5	524,287	5.10E+06	10	26.87
Voting	0	131,071	2.20E+07	168	100
	0.5	131,071	6.60E+06	50	72.58
	1	131,071	2.80E+06	21	56.98
	1.5	131,071	1.80E+06	14	49.28
Waveform	0	4,194,303	1.80E+09	429	100
	0.1	2,753,763	1.00E+08	36	7.62
	0.3	900,948	1.00E+07	11	6.58
	1	167,819	8.00E+05	5	6.18

querying customer data many and many times. Based on the context of analysis and the itemset support values in the customer dataset, officers could allocate economic and structural resources for advertising purposes. For example, advertisements appearing in social events that are most likely to be attended by wealthy people should be targeted to (i) middle age men (support value count: 5.9%+9.8%=15.7%), (ii) young men (3.2%), and (iii) middle age women (2.5%), and (iv) elderly men (1.1%) according to the age distribution in the customer base. On the other hand, elderly wealthy women do not frequently occur in the source data. Hence, it may be not convenient to allocate resources to a relatively small target.

6.3. Effect of the schema-based constraint

Table 6 reports the number of pattern sets mined from the selected UCI datasets by enforcing the schema-based constraint (and neither coverage nor cardinality constraint) as well as the number of corresponding frequent itemsets extracted by enforcing four different minimum support threshold (*minsup*) values. To set the *minsup* values we considered (i) a standard value (1%) common to all datasets, (iii) no threshold value (i.e.,*minsup*=0), to mine the pattern sets including all the possible itemsets, and (ii) two different values per dataset, which depend on the analyzed data distribution.

The maximum number of pattern sets that can be generated from a source dataset is equal to the power set of the number of its frequent itemsets. However, as shown in Table 6, the number of pattern sets mined by enforcing the schema-based constraint is orders of magnitude lower. As discussed in Sections 6.4.1 and 6.4.2, the COPAS algorithm allows us to further reduce the number of mined pattern sets by enforcing the minimum coverage or

the maximum cardinality constraints in association with the schema-based constraint.

6.4. Manageability of the mining results

Result manageability is crucial for data mining applications. The result of the COPAS algorithm consists of a set of pattern sets, which experts may want to manually explore to support decision making. Hence, producing manageable pattern sets is crucial for effectively supporting domain experts during manual result inspection.

The COPAS algorithm generates (at most) one pattern set per schema. Each pattern sets combines all the frequent itemsets characterized by the given schema. In Table 6 we reported the average number of itemsets per pattern set achieved on the UCI datasets. This measure is an indicator of the level of manageability of the mined pattern sets. For example, to perform customer segmentation the average number of itemsets per pattern set indicates the number of distinct segments that analysts should consider once they decide to focus on a specific combination of data facets (see Section 6.2).

The achieved results demonstrate that for most benchmark datasets the mined pattern sets are manageable and thus actionable for performing targeted analyses (e.g., for segmenting customer and planning targeted promotions). For example, when $minsup = 1\%$, the number of average itemsets per group ranges from 5 to 36 itemsets. Hence, each group is, on the average, easily manageable.

To further enhance the manageability of the mining result two complementary strategies have been integrated into the COPAS algorithm: (i) Enforcing a minimum coverage constraint, to prune the pattern sets that do

not cover a significant number of data records (e.g., the facets that represent only a small portion of customers). (ii) Enforcing a maximum cardinality constraint, to prune the pattern sets containing a too large number of itemsets (e.g., the schemata consisting of very large number of segments). Both strategies allow us to prevent the generation of less interesting or potentially unmanageable pattern sets. The effect of these constraints on the characteristics of the mining result is thoroughly discussed in the following sections.

6.4.1. Effect of the minimum coverage constraint

In this section we analyze the impact of the schema-based constraint in association with the minimum coverage constraint (*mincov*).

We run several experiments on the UCI datasets by varying *mincov* between 0 and 100% while enforcing the standard minimum support threshold (i.e., *minsup*=1%). Table 7 summarizes the results achieved on a UCI datasets by setting three different *mincov* thresholds (i.e., 50%, 70%, 90%) and *minsup* = 1% (i.e., the standard minsup value). To gain insights into the achieved results in Figures 1(a), 1(c), and 1(e) we plotted the number of patterns sets by varying the minimum coverage threshold (*mincov*) value and in Figures 1(b), 1(d), and 1(f) we plotted the percentage of pattern sets pruned with respect to the total number of pattern sets that would be generated without enforcing the minimum coverage constraint. Due to the lack of space, Figures 1(a)-(f) refer to a subset of three representative datasets characterized by different data distributions, i.e., Letter-rec. (sparse dataset), Shuttle (fairly dense), and Adult (dense). Similar trends were achieved on the other datasets.

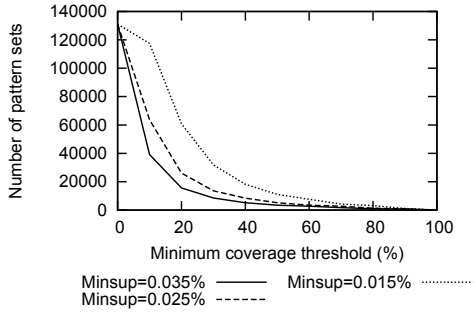
As expected, enforcing the coverage constraint in conjunction with the

Table 7: COPAS. UCI datasets: number of pattern sets and itemsets mined by enforcing different coverage constraint values and *minsup*=1%

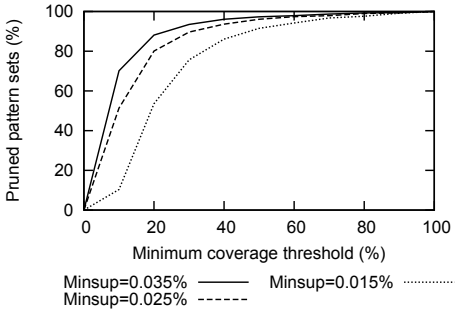
Dataset	mincov (%)	Pattern sets and itemsets satisfying *mincov*				Percentage of pattern sets and itemsets pruned by *mincov*	
		#pat. sets	#itemsets	avg. #itemsets per pat. set	avg. coverage (%) per pat. set	pruned pat. sets (%)	pruned itemsets (%)
Adult	50	6,223	1.18E+05	19	68.4	77.5	63.4
	70	2,549	4.26E+04	17	81.5	90.8	86.8
	90	399	4.42E+03	11	93.9	98.6	98.6
Letter-rec.	50	139	4.08E+03	29	74.9	95	69.8
	70	95	2.72E+03	29	80.1	96.6	79.8
	90	17	1.96E+02	12	98.3	99.4	98.5
Mushroom	50	1,421,841	4.52E+07	32	71.6	67.3	50.5
	70	734,509	2.44E+07	33	82.4	83.1	73.3
	90	157,389	4.47E+06	28	94.0	96.4	95.1
Pendigits	50	155	4.85E+03	31	77.3	97.3	80.5
	70	104	3.20E+03	31	84.2	98.2	87.2
	90	24	4.25E+02	18	97.5	99.6	98.3
Poker	50	76	2.70E+03	36	98.3	0	0
	70	76	2.70E+03	36	98.3	0	0
	90	76	2.70E+03	36	98.3	0	0
Shuttle	50	42	9.16E+02	22	71.5	79	62.9
	70	21	4.35E+02	21	84.8	89.5	82.4
	90	9	1.49E+02	17	94.0	95.5	94.0
Vehicle	50	118,194	3.21E+06	27	64.2	77.5	67.4
	70	32,245	8.73E+05	27	79.3	93.8	91.1
	90	3,329	6.80E+04	20	93.7	99.4	99.3
Voting	50	83,674	1.83E+06	22	66.1	36.2	33.6
	70	28,871	6.34E+05	22	79.7	78	76.9
	90	3,199	4.97E+04	16	93.3	97.6	98.2
Waveform	50	3,731	1.17E+05	31	74.8	97.8	84.8
	70	2,095	6.48E+04	31	87.5	98.8	91.5
	90	939	2.24E+04	24	97.5	99.4	97.1

Table 8: POST-FPMINE. UCI datasets: number of pattern sets and itemsets mined in the stages of POST-FPMINE by enforcing different coverage constraint values and $minsup$=1%

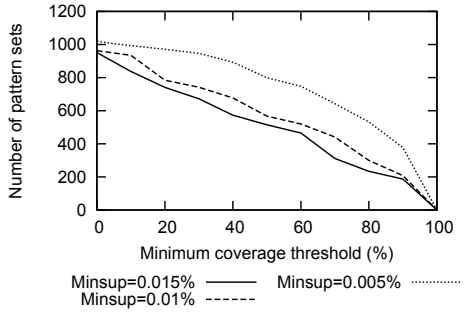
Dataset	mincov (%)	Output of Steps (i)-(ii) of POST-FPMINE Pattern sets and itemsets satisfying schema-based				Output of POST-FPMINE Pattern sets and itemsets satisfying schema-based and *mincov*			
		#pat. sets	#itemsets	avg. #itemsets per pat. set	avg. coverage (%) per pat. set	#pat. sets	#itemsets	avg. #itemsets per pat. set	avg. coverage (%) per pat. set
Adult	50	27,689	3.00E+05	12	28.8	6,223	1.18E+05	19	68.4
	70	27,689	3.00E+05	12	28.8	2,549	4.26E+04	17	81.5
	90	27,689	3.00E+05	12	28.8	399	4.42E+03	11	93.9
Letter-rec.	50	2,794	1.00E+04	5	9.0	139	4.08E+03	29	74.9
	70	2,794	1.00E+04	5	9.0	95	2.72E+03	29	80.1
	90	2,794	1.00E+04	5	9.0	17	1.96E+02	12	98.3
Mushroom	50	4,350,279	9.13E+07	21	41.2	1,421,841	4.52E+07	32	71.6
	70	4,350,279	9.13E+07	21	41.2	734,509	2.44E+07	33	82.4
	90	4,350,279	9.13E+07	21	41.2	157,389	4.47E+06	28	94.0
Pendigits	50	5,798	2.00E+04	4	7.1	155	4.85E+03	31	77.3
	70	5,798	2.00E+04	4	7.1	104	3.20E+03	31	84.2
	90	5,798	2.00E+04	4	7.1	24	4.25E+02	18	97.5
Poker	50	76	2.70E+03	36	98.3	76	2.70E+03	36	98.3
	70	76	2.70E+03	36	98.3	76	2.70E+03	36	98.3
	90	76	2.70E+03	36	98.3	76	2.70E+03	36	98.3
Shuttle	50	200	2.00E+03	12	28.4	42	9.16E+02	22	71.5
	70	200	2.00E+03	12	28.4	21	4.35E+02	21	84.8
	90	200	2.00E+03	12	28.4	9	1.49E+02	17	94.0
Vehicle	50	524,287	9.90E+06	19	36.8	118,194	3.21E+06	27	64.2
	70	524,287	9.90E+06	19	36.8	32,245	8.73E+05	27	79.3
	90	524,287	9.90E+06	19	36.8	3,329	6.80E+04	20	93.7
Voting	50	131,071	2.80E+06	21	57.0	83,674	1.83E+06	22	66.1
	70	131,071	2.80E+06	21	57.0	28,871	6.34E+05	22	79.7
	90	131,071	2.80E+06	21	57.0	3,199	4.97E+04	16	93.3
Waveform	50	167,819	8.00E+05	5	7.6	3,731	1.17E+05	31	74.8
	70	167,819	8.00E+05	5	7.6	2,095	6.48E+04	31	87.5
	90	167,819	8.00E+05	5	7.6	939	2.24E+04	24	97.5



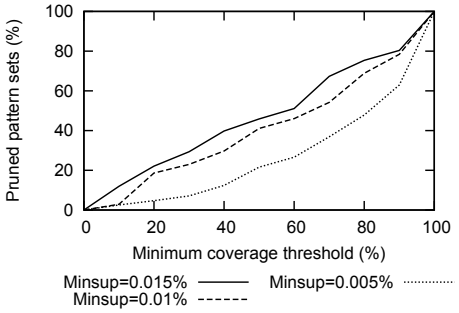
(a) Letter-rec.



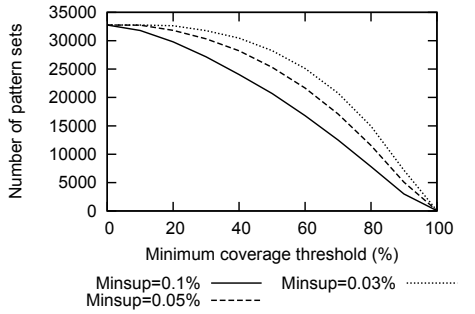
(b) Letter-rec.



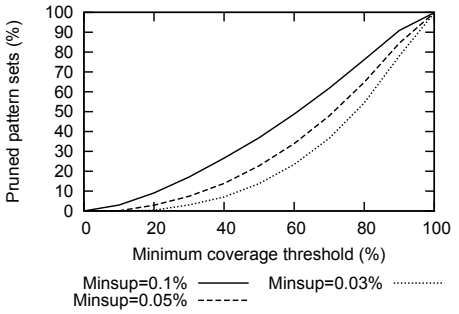
(c) Shuttle



(d) Shuttle



(e) Adult



(f) Adult

Figure 1: Effect of the minimum coverage constraint

schema-based constraint yields a further reduction in the number of pattern sets mined. The selectivity of the *mincov* threshold strictly depends on the analyzed data distribution. Specifically, the selectivity of the constraint is higher when coping with sparser datasets (e.g., Letter), whereas its pruning rate becomes less significant when dealing with denser datasets (e.g., Adult). In sparse (resp. dense) datasets, most itemsets have relatively low (resp. high) support values. Consequently, the coverage of a pattern set, i.e., the count of the number of records covered by any of its itemsets, is usually low (resp. high).

Independently of the analyzed data distribution, the coverage constraint becomes more selective while increasing the *minsup* value, because pattern sets are more likely to contain a fewer number of itemsets and thus their coverage value on average decreases. More specifically, when coping with sparse datasets many itemsets do not satisfy the support threshold. Consequently, the pattern set coverage is on average low and the coverage constraint becomes selective even while setting low support thresholds. For example, more than 50% of the pattern sets mined from Letter-rec. are pruned by enforcing *mincov*=20% (see Figure 1(b)). On the other hand, when coping with relatively dense datasets (e.g., Adult) the selectivity of the coverage constraint becomes significant while enforcing relatively high minimum coverage thresholds (see Figure 1(f)).

Table 7 summarizes the results achieved on all the considered UCI datasets by setting three different *mincov* thresholds (i.e., 50%, 70%, 90%) and $minsup = 1\%$ (i.e., the standard minsup value). The reported results confirm the results obtained on the three representative datasets.

We also analyzed the coverage of the pattern sets generated by the other competitors, i.e., the POST-FPMINE and POST-CPMINE strategies, as intermediate steps. POST-CPMINE never succeeded in generating the candidate pattern sets (see Section 6.1) on all the analyzed datasets due to the combinatorial growth of the number of possibile combinations. POST-FPMINE terminated but it generated a huge amount of (unnecessary) itemsets and pattern sets as intermediate steps. Table 8 (Colums (3)-(6)) reports some statistics on the characteristics of the pattern sets and itemsets generated by POST-FPMINE at the intermediate steps (i) and (ii) (see Section 6.1). Specifically, we analyzed the average coverage per pattern set (Column (6)) and we compared it with those achieved by the COPAS algorithm and reported in Column (6) of Table 7.

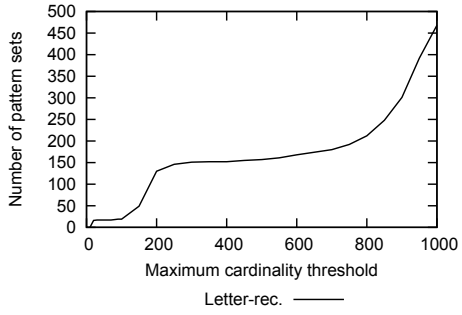
The average coverage of the pattern sets generated by POST-FPMINE as intermediate step is rather low (e.g., 9% for Letter-recognition), because the coverage constraint is not pushed into the mining process. Hence, many potentially uninteresting pattern sets are unnecessarily generated. Furthermore, the number of itemsets and patterns sets generated by POST-FPMINE as intermediate steps is on average at least one order of magnitude higher than the cardinality of the corresponding output sets (i.e., the number of selected itemsets and pattern sets). Hence, the efficiency of the mining process is fairly low. As discussed in Section 6.5, the need for memory-consuming intermediate steps heavily affects the performance of the POST-FPMINE and POST-CPMINE strategies.

Table 9: COPAS. UCI datasets: number of pattern sets and itemsets mined by enforcing different cardinality constraint values and $minsup$=0

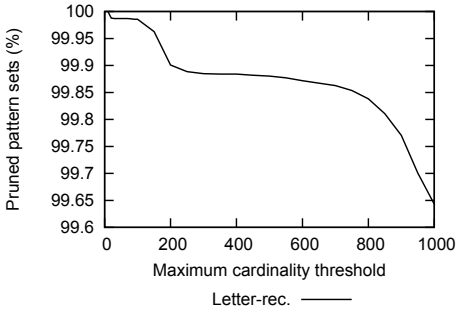
Dataset	maxcard	Pattern sets and itemsets satisfying *maxcard*			Percentage of pattern sets and itemsets pruned by *maxcard*	
		#pat. sets	#itemsets	avg. #itemsets per pat. Set	pruned pat. sets (%)	pruned itemsets (%)
Adult	20	65	7.28E+02	11	99.8	99.9
	30	93	1.47E+03	16	99.7	99.9
	100	331	1.75E+04	53	99.0	99.9
Letter-rec.	20	16	2.57E+02	16	99.9	99.9
	30	17	2.83E+02	17	99.9	99.9
	100	19	4.48E+02	24	99.9	99.9
Mushroom	20	5,353	8.50E+04	16	99.9	99.9
	30	23,403	5.60E+05	24	99.7	99.9
	100	619,529	4.34E+07	70	92.6	96.7
Pendigits	20	17	1.76E+02	10	99.9	99.9
	30	17	1.76E+02	10	99.9	99.9
	100	90	5.69E+03	63	99.9	99.9
Poker	20	21	2.56E+02	12	72.4	99.9
	30	21	2.56E+02	12	72.4	99.9
	100	61	2.40E+03	39	19.7	99.9
Shuttle	20	1	8.00E+00	8	99.9	99.9
	30	1	8.00E+00	8	99.9	99.9
	100	6	3.52E+02	59	99.4	99.9
Vehicle	20	331	4.58E+03	14	99.9	99.9
	30	818	1.72E+04	21	99.8	99.9
	100	14,756	1.06E+06	72	97.2	99.4
Voting	20	413	5.83E+03	14	99.7	99.9
	30	998	2.00E+04	20	99.2	99.9
	100	19,446	1.39E+06	72	85.2	93.8
Waveform	20	259	3.42E+03	13	99.9	99.9
	30	583	1.23E+04	21	99.9	99.9
	100	1,875	1.05E+05	56	99.9	99.9

Table 10: POST-FPMINE. UCI datasets: number of pattern sets and itemsets mined in the stages of POST-FPMINE by enforcing different cardinality constraint values and $minsup$=0

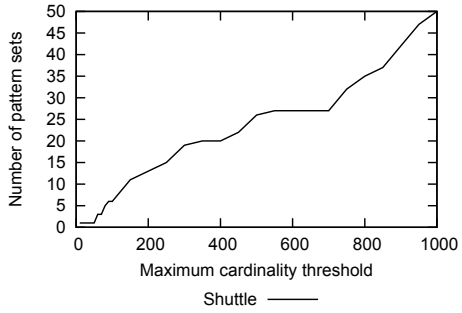
Dataset	maxcard	Output of Steps (i)-(ii) of POST-FPMINE Pattern sets and itemsets satisfying $maxcard$			Output of POST-FPMINE Pattern sets and itemsets satisfying $maxcard$		
		#pat. sets	#itemsets	avg. #itemsets per pat. set	#pat. sets	#itemsets	avg. #itemsets per pat. set
Adult	20	32,767	1.00E+08	3052	65	7.28E+02	11
	30	32,767	1.00E+08	3052	93	1.47E+03	16
	100	32,767	1.00E+08	3052	331	1.75E+04	53
Letter-rec.	20	DNF	DNF	DNF	DNF	DNF	DNF
	30	DNF	DNF	DNF	DNF	DNF	DNF
	100	DNF	DNF	DNF	DNF	DNF	DNF
Mushroom	20	DNF	DNF	DNF	DNF	DNF	DNF
	30	DNF	DNF	DNF	DNF	DNF	DNF
	100	DNF	DNF	DNF	DNF	DNF	DNF
Pendigits	20	131,071	1.10E+09	8392	17	1.76E+02	10
	30	131,071	1.10E+09	8392	17	1.76E+02	10
	100	131,071	1.10E+09	8392	90	5.69E+03	63
Poker	20	2,048	4.00E+08	200317	21	2.56E+02	12
	30	2,048	4.00E+08	200317	21	2.56E+02	12
	100	2,048	4.00E+08	200317	61	2.40E+03	39
Shuttle	20	1,023	2.00E+07	19550	1	8.00E+00	8
	30	1,023	2.00E+07	19550	1	8.00E+00	8
	100	1,023	2.00E+07	19550	6	3.52E+02	59
Vehicle	20	524,287	1.70E+08	324	331	4.58E+03	14
	30	524,287	1.70E+08	324	818	1.72E+04	21
	100	524,287	1.70E+08	324	14,756	1.06E+06	72
Voting	20	131,071	2.20E+07	168	413	5.83E+03	14
	30	131,071	2.20E+07	168	998	2.00E+04	20
	100	131,071	2.20E+07	168	19,446	1.39E+06	72
Waveform	20	DNF	DNF	DNF	DNF	DNF	DNF
	30	DNF	DNF	DNF	DNF	DNF	DNF
	100	DNF	DNF	DNF	DNF	DNF	DNF



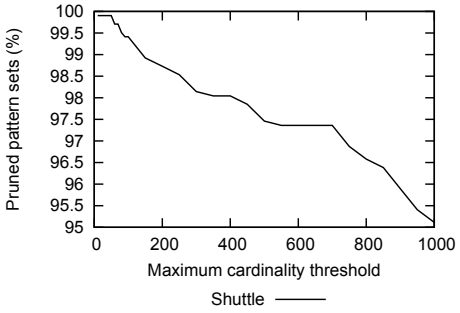
(a) Letter-rec.



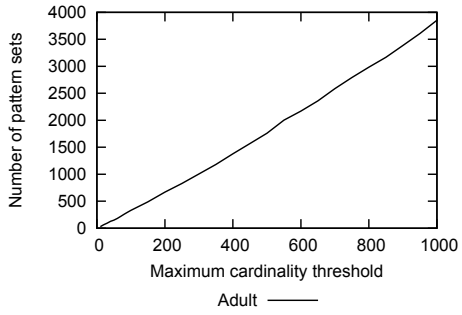
(b) Letter-rec.



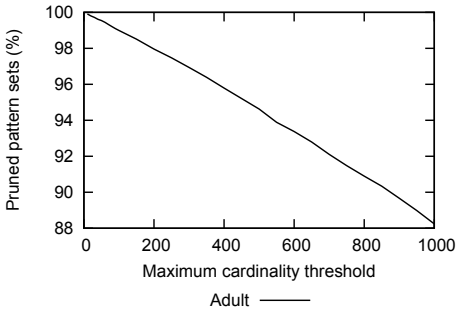
(c) Shuttle



(d) Shuttle



(e) Adult



(f) Adult

Figure 2: Effect of the maximum cardinality threshold

6.4.2. Effect of the maximum cardinality constraint

This section analyzes the selectivity of the schema-based constraint in association with the maximum cardinality constraint (*maxcard*).

We performed several experiments on the UCI datasets by varying *maxcard* between 1 and 10^3 while enforcing no minimum support threshold (i.e., *minsup*=0). Table 9 summarizes the results achieved on a UCI datasets by setting three different *maxcard* thresholds (i.e., 20, 30, 100). Furthermore, Figure 2 plots the number of pattern sets mined from the three representative UCI datasets and the percentage of pruned pattern sets with respect to the total number of frequent pattern sets that would be generated without enforcing the maximum cardinality constraint. Due to the lack of space, in Figure 2 we plotted only the results achieved on three representative datasets with different data distributions, i.e., Letter-rec. (sparse dataset), Shuttle (fairly dense), and Adult (dense). Similar trends were achieved on the other datasets.

For all the analyzed datasets, enforcing the cardinality constraint beyond the schema-based constraint yields a further significant reduction in the number of pattern sets mined (i.e., at least 88% reduction). The selectivity of the cardinality constraint on the number of mined pattern sets is typically higher on datasets with relatively large attribute domains, because their corresponding pattern sets are more likely to contain many itemsets with the same schema. For example, Adult is characterized by relatively small attribute domains (i.e., from 2 to 16 values), whereas Letter-rec. has large attribute domains (i.e., from 16 to 26 values). By comparing the pruning rates achieved on Letter-rec. and Adult (see Figures 2(b) and 2(f)), it turns

out that on Letter-rec. the number of pruned sets remains relatively stable for a relatively large constraint value range, whereas on Adult it decreases roughly linearly. Even though the average attribute domain size of Shuttle is significantly larger than those of Adult (111.6 against 8.1), a relatively small number of itemsets actually occur in Shuttle. Hence, the selectivity of the cardinality constraint is lower than expected.

Table 10 (Colums (3)-(5)) reports some statistics on the results of the intermediate steps performed by the POST-FPMINE strategy (see Section 6.1). Specifically, Column (5) reports the average cardinality per pattern set for all datasets. We compared this result with those achieved by the COPAS algorithm (see Column (5) of Table 7). Since POST-CPMINE never terminated on the analyzed datasets, the corresponding columns were omitted. Note that since we set $minsup$=0 all the selected pattern sets are characterized by coverage equal to 100%.

Even if the outputs of POST-FPMINE and COPAS algorithms are the same, the intermediate steps of POST-FPMINE generated a huge amount of unnecessary itemsets and pattern sets. Specifically, the number of itemsets and pattern set mined by POST-FPMINE as intermediate steps are always at least two orders of magnitude higher than those achieved by the COPAS algorithm. As discussed in Section 6.5, this significantly affects the efficiency of the mining process. For example, on three datasets the POST-FPMINE algorithms was not able to extract the patterns sets and itemsets satisfying both the schema-based and the cardinality/coverage constraint (DNF is reported for those datasets).

Table 11: UCI datasets: execution time of COPAS and POST-FPMINE by enforcing different cardinality constraint values and $minsup$=0

Dataset	maxcard	Execution time (s)	
		COPAS	POST-FPMINE
Adult	20	0.1	50.0
	30	0.1	619.6
	100	0.4	617.6
Letter-rec.	20	0.1	DNF
	30	0.1	DNF
	100	0.1	DNF
Mushroom	20	1.9	DNF
	30	10.3	DNF
	100	679.3	DNF
Poker	20	4.3	1479.6
	30	4.3	1490.8
	100	7.1	1444.4
Pendigits	20	0.1	100.0
	30	0.0	5034.3
	100	0.3	5110.8
Shuttle	20	0.1	50.0
	30	0.1	80.6
	100	0.1	76.8
Vehicle	20	0.1	100.0
	30	0.2	726.9
	100	6.8	715.5
Voting	20	0.0	50.0
	30	0.2	84.1
	100	7.9	90.8
Waveform	20	0.1	DNF
	30	0.3	DNF
	100	1.2	DNF

Table 12: UCI datasets: execution time of COPAS and POST-FPMINE by enforcing different coverage constraint values and $minsup$=1%

Dataset	mincov (%)	Execution time (s)	
		COPAS	POST-FPMINE
Adult	50	3.0	1.4
	70	1.4	1.2
	90	0.3	1.2
Letter-rec.	50	0.3	0.2
	70	0.2	0.2
	90	0.1	0.2
Mushroom	50	1039.5	503.2
	70	589.2	449.9
	90	80.5	406.5
Pendigits	50	0.2	0.2
	70	0.2	0.2
	90	0.1	0.2
Poker	50	7.9	10.0
	70	7.7	9.9
	90	7.7	10.1
Shuttle	50	0.2	0.1
	70	0.1	0.1
	90	0.1	0.2
Vehicle	50	18.7	45.6
	70	5.1	40.8
	90	0.5	37.8
Voting	50	10.3	13.7
	70	3.6	11.2
	90	0.3	9.8
Waveform	50	1.6	2.8
	70	1.0	2.6
	90	0.5	2.6

6.5. Execution time

The goal of this section is twofold. First, it analyzes the execution time spent by the COPAS algorithm on datasets with different characteristics. Secondly, it compares the execution time spent by the COPAS algorithm with those spent by the two competitors described in Section 6.1.

Tables 11-12 summarize the execution times of the COPAS algorithm and the POST-FPMINE strategy achieved on the UCI datasets by enforcing the maximum cardinality constraint and the minimum coverage constraint, respectively. Similar experiments were performed using POST-CPMINE, which never succeeded in extracting all the candidate pattern sets in a reasonable time, i.e., we killed the process after 8 hours. POST-CPMINE did not terminate the extraction process in a reasonable time because in its first step (see Section 6.1) it generates all the combinations of frequent itemsets of arbitrary size. Hence, the number of pattern sets mined by POST-CPMINE at the intermediate Step (i) is equal to $2^{\#\text{ of frequent itemsets}}$. In all the performed expertiments, this number ranges from $2^{(10^3)}$ to $2^{(10^9)}$. Therefore, the task is practically unfeasible in a reasonable amount of time.

The algorithm execution times are inversely correlated with the number of generated pattern sets. The COPAS algorithm appears to be orders of magnitude faster than POST-FPMINE while enforcing the maximum cardinality constraint (see Table 11). Moreover, on three UCI datasets POST-FPMINE does not terminate because of the large amount of (potentially uninteresting) itemsets and pattern sets mined during the first step, which requires too much disk space and main memory.

While enforcing both the minimum coverage and the minimum support

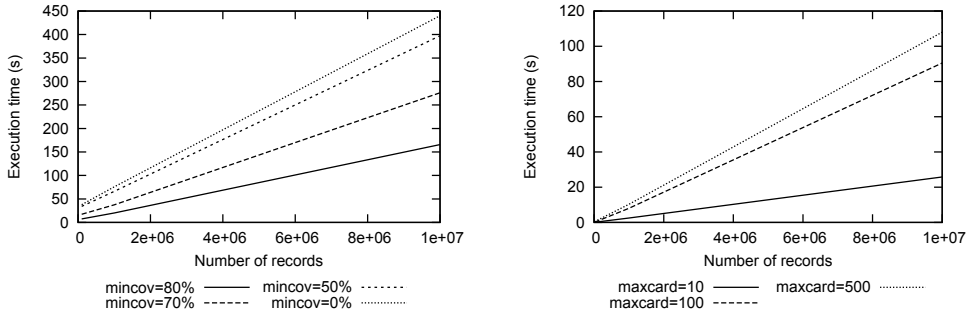
constraints the execution times of the COPAS and POST-FPMINE algorithms are comparable if *mincov* is lower than 90% (i.e., when relative few patten sets are pruned), while COPAS is faster than POST-FPMINE when *mincov* is set to 90% (see Table 12).

6.6. COPAS scalability

We analyzed the scalability of the COPAS algorithm on synthetic data generated by using the generator available at [30]. To perform our analyses we tested synthetic data with different cardinality (i.e., number of records) and dimensionality (i.e., number of attributes). Figures 3-4 summarize the achieved results.

Similar to Apriori [5], COPAS scales linearly with the number of records (see Figure 3). For example, when coping with 10-attribute datasets and by enforcing *mincov*=50% and *minsup*=0.01%, COPAS takes 34s, 66s, and 397s with 10^5, 10^6, and 10^7 records, respectively. Similarly, by enforcing *maxcard*=100 and *minsup*=0 COPAS takes 0.8s, 8s, and 90s with 10^5, 10^6, and 10^7 records.

Because of the non-linear increase in the number of generated combinations, COPAS scales more than linearly with the number of attributes when enforcing either the coverage constraint (see Figure 4(a)) or a relatively high cardinality constraint value (e.g., *maxcard*=100) (see Figure 4(b)). In contrast, when enforcing rather low cardinality constraint values (e.g., *maxcard*=10) COPAS appears to scale approximately linearly. In fact, in the latter case most of the candidate sets are discarded early thus the COPAS execution time is mainly due to I/O operations.



(a) Minimum coverage constraint (minsup=0.01%)

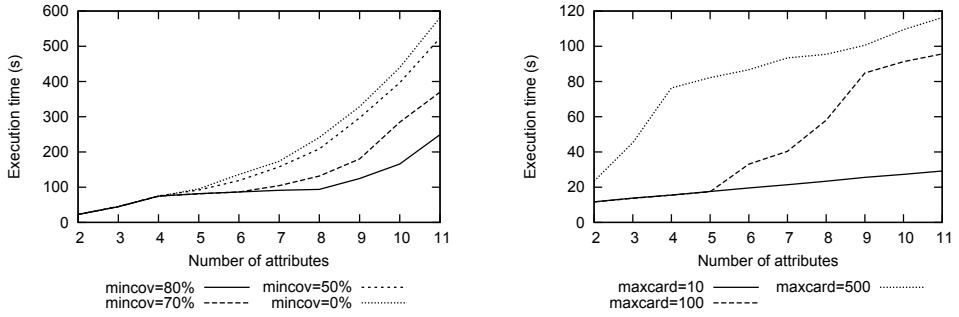
(b) Maximum cardinality constraint (minsup=0)

Figure 3: COPAS scalability with the number of records (number of attributes=10)

7. Conclusions and future work

This paper addresses the pattern set mining problem with global constraints [4]. It presents a new constraint, called schema-based constraints, tailored to relational data. The schema-based constraint exploits the itemset schema to combine all the itemsets that are semantically correlated with each other into a unique pattern set while filtering out the pattern sets covering a mixture of different data facets or giving a partial view of a single data facet. The newly proposed constraint can be efficiently and effectively combined with already existing global constraints. An Apriori-based algorithm to efficiently mine pattern sets under global constraints is also proposed. The experiments demonstrate the selectivity of the proposed constraint as well as the algorithm efficiency and scalability.

As future work, we plan to (i) study the problem of pattern set mining from data equipped with taxonomies by extending existing generalized itemset mining strategies (e.g., [32, 33]), (ii) exploit pattern sets satisfying



(a) Minimum coverage constraint (minsup=0.01%)

(b) Maximum cardinality constraint (minsup=0)

Figure 4: COPAS scalability with the number of attributes (number of records=10^7)

the schema-based constraint to improve the performance of existing itemset-based or associative classifiers (e.g., [34, 35]), (iii) address pattern set mining from quantitative data [26], and (iv) discover interesting groups of infrequent itemsets [27].

References

[1] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, SIGMOD Rec. 22 (1993) 207–216.

[2] D. H. Glass, Confirmation measures of association rule interestingness, Knowledge-Based Systems 44 (2013) 65 – 77.

[3] P.-N. Tan, V. Kumar, Interestingness measures for association patterns: A perspective, KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining (2000).

[4] L. De Raedt, A. Zimmermann, Constraint-based pattern set mining, in: SIAM'07, pp. 1–12.

[5] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: VLDB'94, pp. 487–499.

[6] A. Knobbe, B. Crémilleux, J. Fürnkranz, M. Scholz, From local patterns to global models: The lego approach to data mining, in: ECML/PKDD Workshop, pp. 1–16.

[7] B. Bringmann, A. Zimmermann, The chosen few: On identifying valuable patterns, in: ICDM'07, pp. 63–72.

[8] F. Geerts, B. Goethals, T. Mielikinen, Tiling databases, in: E. Suzuki, S. Arikawa (Eds.), Discovery Science, volume 3245 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2004, pp. 278–289.

[9] T. Guns, S. Nijssen, L. D. Raedt, Evaluating pattern set mining strategies in a constraint programming framework, in: PAKDD'11, pp. 382–394.

[10] A. J. Knobbe, E. K. Y. Ho, Pattern teams, in: PKDD'06, pp. 577–584.

[11] F. Pennerath, A. Napoli, The model of most informative patterns and its application to knowledge extraction from graph databases, in: PKDD'09, pp. 205–220.

[12] J. Vreeken, M. Leeuwen, A. Siebes, Krimp: mining itemsets that compress, Data Mining and Knowledge Discovery 23 (2011) 169–214.

[13] D. Xin, H. Cheng, X. Yan, J. Han, Extracting redundancy-aware top-k patterns, in: KDD'06, pp. 444–453.

[14] I. N. M. Shaharanee, F. Hadzic, T. S. Dillon, Interestingness measures for association rules based on statistical validity, Knowledge-Based Systems 24 (2011) 386 – 392.

[15] T. Guns, S. Nijssen, L. de Raedt, k-pattern set mining under constraints, IEEE TKDE 25 (2013) 402–418.

[16] M. Khiari, P. Boizumault, B. Crémilleux, Constraint programming for mining n-ary patterns, in: CP'10, pp. 552–567.

[17] T. De Bie, Maximum entropy models and subjective interestingness: an application to tiles in binary databases, Data Mining and Knowledge Discovery 23 (2011) 407–446.

[18] B. Goethals, D. Laurent, W. Le Page, C. Dieng, Mining frequent conjunctive queries in relational databases through dependency discovery, Knowledge and Information Systems 33 (2012) 655–684.

[19] C. T. Dieng, T.-Y. Jen, D. Laurent, N. Spyratos, Mining frequent conjunctive queries using functional and inclusion dependencies, The VLDB Journal 22 (2013) 125–150.

[20] S. Ginsburg, S. M. Zaiddan, Properties of functional-dependency families, J. ACM 29 (1982) 678–698.

[21] J. Kivinen, H. Mannila, Approximate inference of functional dependencies from relations, Theor. Comput. Sci. 149 (1995) 129–149.

[22] H. Heikinheimo, E. Hinkkanen, H. Mannila, T. Mielikäinen, J. K. Seppänen, Finding low-entropy sets and trees from binary data, in: KDD'07, pp. 350–359.

[23] R. Agrawal, R. Srikant, Mining association rules with item constraints, in: KDD 1997, pp. 67–73.

[24] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, Generalized association rule mining with constraints, Inf. Sci. 194 (2012) 68–84.

[25] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to data mining, 2005.

[26] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96, ACM, New York, NY, USA, 1996, pp. 1–12.

[27] A. M. Manning, D. J. Haglin, A new algorithm for finding minimal sample uniques for use in statistical disclosure assessment, in: Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 290–297.

[28] A. Frank, A. Asuncion, UCI machine learning repository, 2010. Http://archive.ics.uci.edu/ml.

[29] J. Wang, J. Han, J. Pei, Closet+: searching for the best strategies for mining frequent closed itemsets, in: SIGKDD'03, pp. 236–245.

[30] G. Bruno, L. Cagliero, S. Chiusano, P. Garza, The COPAS algorithm, 2013. Http://dbdmg.polito.it/wordpress/research/copas/.

[31] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: SIGMOD'00, pp. 1–12.

[32] R. Srikant, R. Agrawal, Mining generalized association rules, in: VLDB 1995, pp. 407–419.

[33] L. Cagliero, P. Garza, Itemset generalization with cardinality-based constraints, Inf. Sci. 244 (2013) 161–174.

[34] L. Cagliero, P. Garza, Improving classification models with taxonomy information, Data Knowl. Eng. 86 (2013) 85–101.

[35] E. Baralis, L. Cagliero, P. Garza, Enbay: A novel pattern-based bayesian classifier, IEEE Transactions on Knowledge and Data Engineering 25 (2013) 2780–2795.